REMARKS

Claims 1-12 and 14-18 were pending. Claims 10 and 17 have been cancelled, and claims 1, 14, 16 and 18 have been amended. Accordingly, claims 1-9, 11-12, 14-16, and 18 remain pending subsequent entry of the present amendment.

Prior to providing a claim by claim analysis, the following brief overview of the art cited by the Examiner is presented. The MIPS64 5Kc Processor Core User's Manual, Rev 1.0 (hereinafter "Darren") is directed to an interface that "allows a single coprocessor to be connected to the 5Kc processor core" (page 6-1, line 1). As disclosed on pages 6-1 to 6-2 of Darren, instruction types supported include arithmetic instructions, "[i]nstructions that move data out of the Coprocessor (From COP Ops)", "[i]nstructions that move data into the Coprocessor (To COP Ops)", "[i]nteger instructions that test coprocessor condition bits", and "[c]oprocessor instructions that test integer unit registers". Page 6-14, paragraph 9, line 1 of Darren discloses "The 5Kc will never dispatch more than one instruction per cycle." Additionally, Darren discloses: "an arithmetic coprocessor instruction can be dispatched at the same time as a To COP Op or From COP Op." (page 6-14, paragraph 7, lines 1-2). Accordingly, Darren discloses more than one coprocessor instruction per cycle may be dispatched, but the simultaneously dispatched coprocessor instructions are of different types. In contrast, Applicant's presently claimed invention recites more than one coprocessor instruction per cycle to be dispatched of the same type.

35 U.S.C. § 102 Rejections

In the present Office Action, claims 1-12 and 14-18 stand rejected under 35 U.S.C. § 102 (b and f) as being unpatentable over Darren. However, Applicant submits each of the pending claims recite features not disclosed by the cited art. Accordingly, Applicant traverses the above rejections and requests reconsideration.

Currently amended independent claim 1 recites a coprocessor interface, which includes:

> "an instruction transfer signal group for transferring a plurality of instruction types from the CPU to the coprocessor; and
>
> a busy signal group, coupled to said instruction transfer signal group, for signaling said instruction transfer signal group when one or more of said plurality of instruction types cannot be transferred;
>
> wherein when said instruction transfer signal group receives said signaling for one or more of said plurality of instruction types, said instruction transfer signal group does not transfer those instruction types, but transfers other ones of said plurality of instruction types;
>
> wherein said instruction transfer signal group is configured to transfer two or more of said plurality of instruction types from the CPU to the coprocessor, in parallel;
>
> wherein said two or more of said plurality of instruction types that are transferred in parallel may be of a same type or of a different type."

Therefore, claim 1 recites "said two or more of said plurality of instruction types that are transferred in parallel may be of a same type." In contrast, Applicant has reviewed the cited reference and submits Darren does not disclose at least these features.

In the cited reference, there are various supported types of coprocessor instructions disclosed on pages 6-1 to 6-2 of Darren. First, Darren teaches that the interface cannot transfer two data movement instruction types, To COP Op and From COP Op, from the CPU to the coprocessor, in parallel. The two signals, CP_ts and CP_fs, are described on page 6-4, Table 6-2 of Darren as being control signals asserted "after a To coprocessor data movement instruction is available" and "after a From coprocessor data movement instruction is available", respectively. Darren discloses "CP_ts and CP_fs will never be asserted at the same time." (page 6-14, paragraph 7, line 1). Table 6-2 on page 6-4 of Darren reiterates that these two signals "will never be asserted in the same cycle".

Second, Darren teaches that the interface cannot transfer two arithmetic instruction types from the CPU to the coprocessor, in parallel. The signal, CP_irs, is described on page 6-3, Table 6-2 of Darren as being a control signal asserted "the cycle after an arithmetic coprocessor instruction is available". Applicant has reviewed the cited reference and submits this is the only disclosed control signal asserted "the cycle after an arithmetic coprocessor instruction is available". Thus, only one arithmetic coprocessor instruction may be dispatched at the same time. Page 6-15, Figure 6-1 of Darren discloses that only one arithmetic instruction is dispatched at the same time.

Third, Darren teaches that the interface cannot transfer two "integer instructions that test coprocessor condition bits" from the CPU to the coprocessor, in parallel. These instructions are not transferred from the CPU to the coprocessor. Table 6-2 on page 6-7 of Darren shows the coprocessor condition signals are all inputs to the CPU. There are no signals from the CPU to the coprocessor. Thus, the "integer instructions that test coprocessor condition bits" are executed outside of the coprocessor. Furthermore, Darren discloses on page 6-18, paragraph 6, line 1: "[f]or every arithmetic and To/From instruction dispatched, the full set of 8 condition code bits is transferred back to the integer unit as they would be after the instruction completes." From the above, the interface cannot transfer two arithmetic or two To/From instructions from the CPU to the coprocessor, in parallel. Thus, the "integer instructions that test coprocessor condition bits" cannot be executed in parallel. The interface merely signals the CPU "when the coprocessor condition bits are available" and of the valid values of the coprocessor condition bits when they are available. Figure 6-4 on page 6-19 of Darren shows an example of the waveforms for testing the coprocessor condition bits. Besides the arithmetic instructions in Figure 6-4, there are no signals transferred from the CPU to the coprocessor. The interface does not transfer "integer instructions that test coprocessor condition bits" from the CPU to the coprocessor, in parallel.

Fourth, Darren teaches that the interface cannot transfer two "coprocessor instructions that test integer unit registers" from the CPU to the coprocessor, in parallel. For example, Darren discloses:

> "For two special arithmetic coprocessor instructions, the integer unit transfers this information. These instructions are MOVN.fmt and MOVZ.fmt. When these instructions are dispatched to the coprocessor, they are also dispatched to the integer pipeline. In this way, the integer unit can test if RT==0 or not." (page 6-20, paragraph 5, lines 1-3)

Above, it was shown the interface cannot transfer two arithmetic instruction types from the CPU to the coprocessor, in parallel. Since the "coprocessor instructions that test integer unit registers" are "two special arithmetic coprocessor instructions", then the interface cannot transfer two "coprocessor instructions that test integer unit registers" from the CPU to the coprocessor, in parallel.

Applicant believes the features of claim 1 to be patentably distinguished from the cited art for at least the above reasons. As independent claims 14 and 16 include features similar to that of claim 1, each of these claims are patentably distinct for at least these similar reasons. As each of the dependent claims 2-9, 11-12, 15 and 18 includes the features of the independent claims on which they depend, each of the dependent claims are patentably distinct for at least the above reasons as well.

In addition to the above, the cited reference was authored by one of the inventors (Darren Jones) of the present application on behalf of the inventors of the present application. However, Darren Jones is not the sole inventor of the presently claimed invention. Rather, the inventorship is as stated in the declaration filed with the present application.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

Respectfully submitted,

/James W. Huffman/

James W. Huffman
Reg. No. 35.549
ATTORNEY FOR APPLICANT(S)

Date: 12/16/2006